

Bridging Versioning and Adaptive Hypermedia in the Dynamic Web

Evgeny Knutov, Mykola Pechenizkiy, Paul De Bra

Eindhoven University of Technology, Department of Computer Science
PO Box 513, NL 5600 MB Eindhoven, the Netherlands
{e.knutov, m.pechenizkiy}@tue.nl, debra@win.tue.nl

Abstract. Web Dynamics has been recently considered in the context of the analysis of how people search and re-search information on the web. There are lots of challenges and opportunities when considering user behaviour. In this paper we propose the way to tackle some of them by applying versioning methodologies (as a backend solution) in the context of content changes, user re-visitations and re-searches on the web, as well as Adaptive Hypermedia (AH) techniques to overcome visualisation issues (as a frontend solution). Essentially we bridge versioning and AH in the field of Web Dynamics showing how versioning helps to make the adaptation scrutable.

Keywords: adaptive hypermedia, versioning, web dynamics, user modelling.

1 Introduction and Background

In a dynamic information environment such as the Web the understanding of the constantly evolving content is becoming an issue [1]. The main problem is that user behaviour is influenced by the fact that changes to the content are observed. These changes cause re-visitation and re-searches for the “same” information. There is a number of possible ways to make users more comfortable with changing information, such as showing changes (e.g. in additional snippets while presenting search results), or by providing more general browsing or hierarchical views of information being explored by the user.

Versioning (also known as source or revision control or code management) is the management of multiple revisions of the same instance of information. It is commonly used in software development to support the team work and the sustainability of a product. Independent of how “versions” of a Web page are really created or generated, changes in retrieved information units can be identified by incrementing an associated version and binding it historically to a user.

In order to cope with the information overload issues, changes and evolving structures on the web we may think of using versioning approaches and technologies. Besides this, the rates of web re-visitations and re-searches (which are estimated to be very high [1, 6], e.g. re-visitation rates are about 50-80%) incite us to investigate versioning in the area of user visitation and search behaviour.

In this paper we essentially consider a number of scenarios of bridging versioning methodologies with AH. The first one is the backend solution, which employs conventional source control approaches (such as used in the software development process or content management systems). Secondly, we consider the front-end solution that uses AH methods and techniques, presenting versioned information to the user. And as a result we think that these two can be merged moving towards an intelligent versioning cache system or a proxy that would serve as an Adaptive Hypermedia System (AHS) and would function as a proxy between conventional web information system and a user (user profile) retrieving, keeping track of, dispatching and presenting adapted information corresponding to the versions of the evolving environment (web content, user queries, sites visited).

Our goal in this paper is to show how versioning may help to scrutinize adaptation. We consider backend solutions in terms of versioning methodologies and techniques (section 2). We touch upon frontend issues involving an AH toolkit (methods and techniques) as the way to present versioned changes (section 3). As a proof of concept we present a use-case of the user search and re-visitation activities on the web (section 4). And we conclude the paper describing innovative aspects and prospective solutions that versioning can provide (section 5).

2 Versioning to Support AH in a Dynamic Web Environment

Considering the basic concepts of versioning we can come up with the following classes of operations which will reflect typological and structural types of changes and capture changes in the dynamic web environment. The following taxonomy of changes was mentioned in [5] in application to ontology evolution, and extended in the field of AH [7]. Here we extend and elaborate it in terms of Web Dynamics, describing properties and potentials of versioning operation which could be used mostly as a backend solution to store, keep track of changes and retrieve them for further analysis:

Transformation – is a set of actual changes that take place over the evolving structures, content, properties, and information instances in general. For instance these changes can be found at the DOM level of a web page in DiffIE [2] (here changes are identified comparing hash trees of the pages). These changes include: *Addition, Change, Deletion and Movement*.

Conceptual changes – (include concept and concept relationship changes) refer to conceptual and more abstract (than real transformation) changes of the structure, relationships (including relations between constructs of two versions of a model or a system), or presentation aspects. This type of changes includes changes of types, relations, conceptual representation of the knowledge, system notions and definitions.

Descriptive changes – dealing with metadata describing the intentions, user or author credentials, and reasoning behind the change or operation (information access, content retrieval, etc.). Descriptive changes can't contribute to the actual transformation of a data instance (e.g. change User Model (UM) attributes, access to the web page), however they facilitate reasoning over multiple versions of the same instance (e.g. taking into account contextual information about the time or place of

change in order to refine user query). Descriptive changes could be exploited to describe the provenance of information, thus introducing even more elaborated capabilities changes, which can be used to overcome information reliability and currency issues, as well as to provide additional explanations to support complex reasoning.

Context changes – describe the environment in which the current update occurs and the environment where it is valid, e.g. changing a particular web page, concept in the Domain Model (DM) in AHS or the hyperlink from one page to another. Context changes to some extent are the changes that drive versioning in most cases, either considering usage or user context (e.g. time changes). At the same time the system environment can be considered also as a context. Context changes are usually the most space/time and analysis effort consuming. They usually require domain experts to take part in the analysis in order to capture the complete picture of a particular change and be able to support complex cases of reasoning and draw the right conclusions.

In [4] we can see that to overcome the overload issue so-called “Milestones in Time” (aka. “Memory landmarks”) are described. This is a set of public and personal time labels referring to a particular event, (in fact timestamp) according to which the information can be easily retrieved. This mostly relies on the visualisation of the results on the personal content index while displaying the results of queries.

The SIS (“Stuff I’ve Seen”) system [6] is a personal indexing and search system which provides personalised result for the aforementioned “Memory landmarks” visualisation. SIS indexes full text and metadata which enables a fast way to search through the user content. The result in SIS was obvious (and has a psychological aspect of an episodic memory) – such an arrangement reduces the search time and in some cases influences whether the user would prefer to re-search information or go directly to an already defined public or personal “landmark”.

3 AH Presentation Aspects

There has been much stress placed on the visualization part of re-searches and re-visitation cases, presenting mostly ‘historical’ snippets in the search result list or browser extensions (e.g. DiffIE) which highlighted the difference in the newer versions of the document.

Here we propose to have a look at the visualization issues throughout AH research and use content and presentation adaptation techniques introduced there in order to support versioning visualisation. At the same time one may consider different versions of some content and apply the same techniques from AH field taking into account the fact that different version just represent the different variants of the same fragment as it is done in AHS.

Adaptation techniques and methods refer to methods of providing adaptation and their generalization correspondingly. Techniques of an implementation part are usually characterized by a specific approach or algorithm. This set of techniques (together with methods generalising them) comprises a toolkit of AH. Both techniques and methods can be applied to content, presentation and navigation

adaptation. We believe that presentation techniques including content and layout adaptation can be used with slight modifications according to versioning needs in enhancing the ‘historical difference’ experience of the end-user.

Content adaptation techniques: the presentation of changed information can be influenced essentially in two ways: by showing/hiding or by emphasizing/deemphasizing it. Through inserting, removing, or altering fragments the representation of a current version of the information content is changed. Other techniques: dimming, sorting, zooming, and stretchtext may keep the old information available.

Presentation adaptation techniques: as shown in AH - changing the presentation can be used to either (de)emphasize parts of the content or to suggest links. Changed parts of information may also be presented within a predefined template layout which is reflected in presentation specification (e.g. presenting the differences using predefined CSS templates).

4 Use-Case: User Web Activity in a Versioned Environment

Here we present a use-case bridging versioning approaches used to track user activities in the dynamic web environment with the AH presentation of the versioned content (e.g. user search results history, changes in web pages, etc.).

Here we have a user, who searches and browses through the web. The initial state of UM (UM.ver.1) starts accumulating search and visitation history, hierarchically structuring UM versioned instances according to descriptive changes, context (or some other changes) that have happened. After a few searches there is already enough

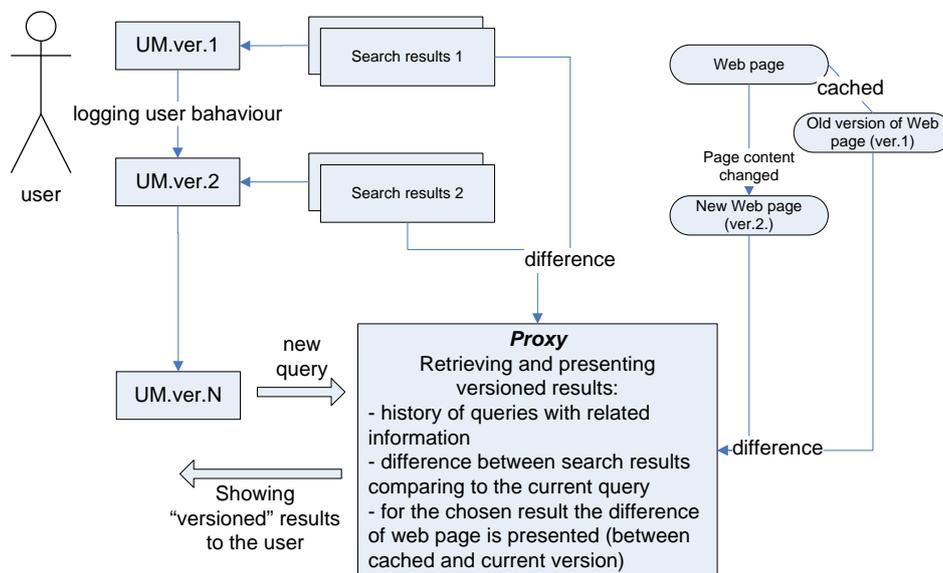


Fig. 1 User web activities in ‘versioned’ environment

information gathered to consider. As the user continues the interaction and posts a new query. A Proxy retrieves the previous state of UM and information such as search queries and corresponding result lists with (descriptive, context, conceptual (if any available)) information about changes to UM. These changes are processed and compared by the proxy and then presented to the user, providing an opportunity to see what other related queries and results he has already done. As a result actual changes of a particular page (chosen by the user from his search results) can be retrieved and presented using AH techniques (e.g. annotating some new part of the page with old descriptions or showing both versions on one page and dimming the old content). Search results could be also presented using AH techniques and re-arranged according to a new global or personalized ranking that may change over time for various reasons (e.g. in e-learning system certain information object has changes a status from 'to be learnt' to 'mastered' or in the news search particular event is not a news any more or with respect to a user this news is already known, etc.).

Here the difference with other users' results may be presented as well when considering UM versions of multiple users from similar group or random user with the similar query. Furthermore as mentioned in [7] this comparison can be used not only to present information differences but to make recommendations and suggestions as well.

5 Innovative Aspects and Prospective Solutions

Here we would like to conclude and outline the advantageous, innovative aspects and prospective versioning solutions in the web dynamics. These are the following:

System Design and Authoring – versioning helps to create, maintain, re-use, reconcile concurrent versions of an application, model or a particular property or value (e.g. web site map and the corresponding content, previously discovered web searches and new pattern suggestions), saving authoring and design effort.

Storing – versioning provides an efficient way to store changes, label and annotate them. It facilitates convenient, hierarchical structure presentations and offers a number of operations to handle it. This saves space for a large scale systems and keeps all the historical changes in order as well as provides a convenient way to retrieve these changes for further reasoning or analysis.

System Maintenance and Support – structured changes and a number of operations (e.g. merge, resolve, commit, tag/label, head, branching, etc.) are sufficient to maintain and reconcile application conflicts, support concurrent versions, inherit functionality between system versions, or resolve functionality conflicts.

Logging – logging changes provides flexible playback (roll-back) possibilities and can serve as a basis for system analysis and evaluation. Logging in terms of user profile updates will provide a ground for user comparison (e.g. comparing behaviour patterns in order to provide suggestions or recommendations). A representative example of logging capabilities that can be done with the following types of changes at the DOM level of a web page (addition, deletion, movement, change) is shown in [2]. Maintaining versioned log structures helps with noisy data and exploits structural tracking of changes.

Analysis – step-by-step system and user behaviour analysis is facilitated. Tagging and labelling can be used for more complex analysis such as OLAP. Versioning which provides hierarchical incremental logs exploits the transparency of the system functionality and evolution of the user web environment (such as labelling and tagging in the version control system can be used to re-construct user behavior and dynamics of visitations and searches, and at the same time to provide suggesting and recommendations while comparing these results to other users with similar properties).

6 Conclusions and Future Work

In this paper we mapped conventional versioning approaches onto the field of Adaptive Hypermedia in the Dynamic Web environment. We presented the taxonomy of versioning changes in the context of Web Dynamics and revisited AH presentation techniques that can help to overcome visualization issues. As a result we essentially bridge versioning and AH in dynamic web environment, presenting backend and frontend solutions, showing advantages of this approach and outlining perspectives.

As a further work we consider describing versioning methodologies for an AH framework, devising universal techniques and investigating versioning technologies (e.g. source control, historical data bases, etc.) that can entirely meet the requirements of the adaptation framework.

Acknowledgements. This research is supported by NWO through the “GAF: *Generic Adaptation Framework*” project.

References

- 1 Adar, E., Teevan, J., Dumais, S. T., Elsas, J. L.: The web changes everything: understanding the dynamics of web content. In *Proc. of 2nd ACM Int. Conf. on Web Search and Data Mining (WSDM'09)*, pp. 282-291. ACM, New York, NY, (2009)
- 2 Teevan, J., Dumais, S. T., Liebling, D. J., Hughes, R. L.: Changing how people view changes on the web. In *Proc. of 22nd ACM Symposium on User Interface Software and Technology (UIST '09)*, pp. 237-246. ACM, New York, NY (2009)
- 3 Griffiths, J., Millard, D. E., Davis, H., Michaelides, D. T., Weal, M. J.: Reconciling Versioning and Context in Hypermedia Structure Servers. In: *Proc. Metainformatics Int. Symp., (MIS 2002)*. pp. 118-131. Springer, Heidelberg (2002)
- 4 Ringel, M., Cutrell, E., Dumais, S., Horvitz, E.: Milestones in time: The value of landmarks in retrieving information from personal stores. In *Proc. of Interact (2003)*
- 5 Klein, M., Fensel, D., Kiryakov, A., Ognyanov, D.: Ontology versioning and change detection on the Web. In *Proc. of 13th Int. Conf. on Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, pp. 247-259. Springer, Heidelberg (2002)
- 6 Dumais, S. T., Cutrell, E., Cadiz, J. J., Jancke, G., Sarin, R., Robbins, D.: Stuff I've Seen: A system for personal information retrieval and re-use. In *Proc. of SIGIR'03*, pp. 72-79 (2003)
- 7 Knutov, E., De Bra, P., Pechenizkiy, M., Versioning in Adaptive Hypermedia, In *Proc. of Workshop on Dynamic and Adaptive Hypertext: Generic Frameworks, Approaches and Techniques*, pp. 61-71. CEUR-WS.org, Aachen, (2009)